F.Spagna 2 maggio 97

NOTE SUI
SISTEMI (TECNOLOGIE/AMBIENTI) DI OGGETTI DISTRIBUITI
PER LO SVILUPPO DI APPLICAZIONI DISTRIBUITE

MODELLI A COMPONENTI SOFTWARE E OGGETTI DISTRIBUITI

I sistemi di oggetti distribuiti (D.O.S.: Distributed Object System) sono tecnologie che permettono a componenti software (oggetti) di comunicare tra di loro direttamente su reti in modo affidabile, sicuro ed efficiente.

Un modello a componenti software e' una specifica su come sviluppare componenti software riutilizzabili e su come questi oggetti componenti possono comunicare tra di loro. Gli sviluppatori utilizzatori di componenti software devono trattare dettagli a basso livello di programmazione per l'implementazione del componente, ma gli utilizzatori dei componenti stessi possono metterli insieme in modo semplice (ad esempio in un ambiente visuale).

Con questa tecnologia le applicazioni o delle combinazioni di applicazioni e dati possono invocarsi ed incapsularsi l'un l'altra (piu' applicazioni esistenti, con i loro dati, possono essere incapsulate ed integrate in una nuova applicazione costruita piu' grande).

I socket di comunicazione agli oggetti sono incapsulati dal D.O.S., togliendo questa complessita' al programmatore (si permette agli sviluppatori di scrivere ai servizi di comunicazione piuttosto che costruire direttamente ogni volta la connettivita' dentro le applicazioni). I programmatori possono usare oggetti presenti su sistemi remoti come se fossero locali.

Le applicazioni costituite da componenti sono costruite mettendo insieme un certo numero di componenti indipendenti tramite codice scritto in un qualche linguaggio di programmazione. Tale codice rappresenta cio' che fa stare insieme i componenti ed agisce sulle proprieta' di essi o ne invoca i metodi, rispondendo agli eventi. Esistono oggi molti diversi modelli di componenti.

Un cliente accede ad un oggetto rivolgendogli una richiesta (evento).

L'IDL (Interface Definition Language) e' il linguaggio che specifica le interfacce tra oggetti.

COMPONENTI SOFTWARE

Gli oggetti (come DCOM e DSOM) permettono agli sviluppatori di creare oggetti (cioe' insiemi di dati ed operazioni che possono essere esguite sui dati stessi), scritti in qualsiasi linguaggio, su cui possono operare altri programmi o oggetti a un livello binario standard.

I componenti software devono rispettare delle convenzioni cosi' che gli ambienti di sviluppo delle applicazioni possano manipolarli.

Un componente e' un oggetto software che ha i seguenti elementi:

- proprieta' caratterizzate da valori leggibili e/o scrivibili da un programma
- metodi, cioe' azioni invocabili a run-time da un programma

- eventi, cioe' notificazioni in risposta ad un cambiamento di stato dell'oggetto.

Le caratteristiche che devono avere i componentti per funzionare sono:

- devono essere capaci di autodescriversi dal punto di vista di proprieta' ed eventi,
- devono permettere un editing grafico delle loro proprieta' (ambienti visuali),
- devono poter essere definiti da un linguaggio di programmazione (ambienti non visuali),
- devono poter generare eventi che li legano ad altri componenti.

I componenti sono opachi in quanto la loro implemntazione interna e' nascosta all'esterno (possono essere disponibili anche soltanto in forma binaria).

ORB (Object Request Brokers)

Costituiscono il fondamento di ogni Object Management Architecture (OMA) e gestisono tutte le comunicazioni tra i loro componenti, identificando e localizzando gli oggetti e permettendo ad essi di interagire in un ambiente distribuito eterogeneo indipendentemente dalle piattaforme sui quali gli oggetti risiedono e dalle tecniche usate per implementarli. Un ORB assolve il suo compito per mezzo degli Object Services che possono creare oggetti, controllarne l'accesso, conservare traccia degli oggetti rilocati, etc. L'utilizzatore ha a disposizione le Common Facilities e gli Application Objects per utilizzare questi servizi. La funzionalita' principale di un ORB e' quella di passare le richieste dei clienti alle implementazioni degli oggetti sulle quali esse sono invocate. Cio' avviene attraverso uno stub, che e' la mappatura tra il linguaggio usato dal cliente (qualunque esso sia purche' sia supportato), ed il core dell'ORB, che trasferisce la richiesta all'oggetto, il quale riceve la richiesta tramite uno skeleton. Le comunicazioni tra l'implementazione di un oggetto ed il core ORB sono effettuate dall'Object Adapter (OA) che gestisce i servizi relativi. Ogni sistema specifico (ad esempio database) richiede un suo proprio object adapter.

SCELTA DI UN ORB

Le cose stanno evolvendo velocemente e gli statement fatti finora vanno rivisti, cosi' per quanto riguarda la limitazione di DCOM all'utilizzazione su piattaforme Windows (Il porting su altre piattaforme e' in corso), come la non disponibilita' di sistemi ORB di tipo CORBA. E la crescente incredibile popolarita' di Java si inserisce nel quadro.

LINGUAGGIO IDL

Un compilatore IDL processa le istruzioni del linguaggio IDL. Esiste un CORBA IDL. Il linguaggio OMG IDL separa le speciche di interfaccia dalle implementazioni e puo' essere mappato su vari linguaggi di programmazione tra cui il C, il C++, Ada e Smalltalk. E' in corso l'estensione per Java (Java ORB): vi stanno lavorando HP ed Oracle.

ARCHITETTURA DCOM (Distributed Component Object Model)

COM (Component Object Model) e' il predecessore di DCOM.

DCOM e' un protocollo che permettono a componenti software (oggetti) di comunicare tra di loro direttamente su una rete (compreso HTTP di Internet) in modo affidabile, sicuro ed efficiente

Caratteristiche:

- disponibile in versione beta solo dal 1996 (immaturo)
- sviluppato da Microsoft (proprietario)
- per piattaforme Windows potrebbe essere una buona scelta

- la specifica COM (Microsoft e DEC) stabilisce l'implementazione di OLE su computer non-PC

(CORBA e COM potrebbero un giorno confluire o coesistere interoperativamente)

- difficile la multipiattaforma (IDL/ODL)

Microsoft ha sottoposto DCOM per il processo di standardizzazione a The Open Group (TOG), formatosi nel 1996 per concentrazione tra due consorzi per i sistemi aperti: l'X/Open e Open Software Foundation (OSF).

La Software AG tedesca lavora sul "porting" di DCOM su varie piattaforme non-Windows (Solaris, HP-UX, MVS, Linux) e Digital su Digital Unix e Open VMS.

OLE (Object Linking and Embedding) di Microsoft usa DCOM per creare documenti composti.

ACTIVEX

Active X e' un'estensione di COM, con qualche caratteristica aggiunta per il lavoro in rete. Si tratta di un'architettura a componenti adatta per tutti i linguaggi (i controlli ActiveX possono essere sviluppati in diversi linguaggi di programmazione, Java compreso).

I controlli ActiveX comunicano anche se posti su macchine diverse (reti eterogenee) attraverso il meccanismo a basso livello DCOM che permette loro, in relazione ad eventi, l'accesso ai metodi esposti degli altri controlli, che devono avere come argomenti l'insieme di parametri (signature) rappresentant i i dati che devono essere passati.

La tecnologia ActiveX rappresenta un avanzamento di OLE, che permette l'inserimento dei controlli in pagine Web e la loro risposta interattiva agli eventi.

Lo sviluppo di controlli ActiveX e' praticamente impossibile senza un ambiente di sviluppo avanzato che automatizzi il lavoro, in quanto il lavoro sottostante e' complesso.

I controlli ActiveX non hanno per la sicurezza lo stesso approccio di Java e possono accedere liberamente alle risorse del computer (il che conferisce loro una grande potenza, molto utile ad esempio nelle applicazioni intranet). Il meccanismo di sicurezza prevede un controllo di una firma elettronica, ma non e' sufficiente per garantire la sicurezza: la mancanza di una sicurezza intrinseca lascia un grosso buco dal punto di vista della safety e della security. Laddove considerazioni di sicurezza prevalgono non e' utilizzabile.

ActiveX non e' cross-platform.

Microsoft ha sviluppato la possibilita' di invocare i controlli ActiveX da Java e Java dai controlli ActiveX.

ARCHITETTURA CORBA (Common Object Request Broker Architecture)

CORBA e' la specifica di un sistema che permette l'interoperabilita' tra oggetti in un ambiente eterogeneo distribuito, in modo trasparente per il programmatore.

Caratteristiche:

- esiste dal 1990: e' piu' maturo (e' alla sua 3a generazione)

- supportato da una ventina di produttori con molteplici implemenazioni

- supporta Windows ed altre piattaforme (e' indipendente dalla piattaforma: multi-CPU/OS/linguaggio)

- permette di collegare Windows con Unix

- piu' ricco e piu' robusto di DCOM

- difficile la multipiattaforma (IDL)

E' stato sottoposto all'OMG (Object Management Group), consorzio non-profit creato nel 1989 (oltre 500 tra vendors, sviluppatori e utilizzatori rappresentati) con lo scopo di promuovere teoria and pratica della ttecnologia degli oggetti in sistemi di computer distribuiti. Tale organismo sta lavorando per la creazione di standard industriali aperti per l'interoperabilita' e portabilita' delle applicazioni orientate ad oggetti distribuiti.

Gli oggetti in CORBA possono essere oggetti della stessa applicazione, programmi separati, oggetti di database ed altro.. Essi possono essere scritti in una varieta' di linguaggi di programmazione.

Ha gia' servizi sviluppati: COS (Common Object Services), Naming, Events, Security.

ARCHITETTURA SOM E DSOM

SOM (System Object Model)

Il modello di oggetti SOM (System Object Model) di IBM (di cui DSOM e' il subset distribuito per lavoro in rete) e' aderente alle specifiche CORBA.

SOM e' un modo per i programmi di riusare componenti binari scritti in linguaggi diversi mediante un modello di oggetti indipendenti dal linguaggio. E' un ORB compatibile con CORBA, che ha rappresentato una risposta a OLE/COM.

Finora usato quasi esclusivamente da IBM, ma IBM spera di allargarne l'uso. Inizialmente introdotto nella Workplace Shell di OS/2, per risolvere i problemi di incompatibilita' di linguaggio ed incompatibilita' binaria tra classi nella programmazione object oriented (OO), adesso e' disponbile su diversi sistemi operativi tra cui AIX e Windows (IBM sta abbandonando la politica di ignorare Windows).

SOM lavora dentro un singolo spazio di memoria. DSOM e' il sottoinsieme distribuito di SOM per lavorare in una rete su macchine remote, rispondente lle specifiche CORBA OMG, con una o due eccezioni importanti erditate da SOM (framework di metaclassi e possibilita' di cambiare la definizione di calssi senza dover ricompilare i programmi che le usano).

Il prodotto che incorpora SOM e' conosciuto come SOMobjects.

Un oggetto SOM puo' essere reso persistente mediante il framework PSOM (Persistence SOM).

DSOM

- e' un modello di oggetti indipendenti dal linguaggio

- e' un modo con cui i programmi possono riutilizzare componenti binari scritti in diversi linguaggi.

- disponibile su un certo numero di sistemi operativi di IBM (tra cui AIX e Windowa).

- e' il piu' diffuso ORB (Object Request Broker)

- e' uno standard industriale (non solo IBM): va su una varieta' di piattaforme

- ha alcune proprieta' ereditate da SOM non ancora presenti in CORBA (estensioni proprietarie)

- e' indipendente dal linguaggio: il supporto per C e C++ e' gia' implementato ed e' previsto quello per Cobol, Smalltalk, Ada ed altri linguaggi (ultimamente HP e Oracle stanno

lavorando su Java)

OPENDOC

Lo standard OpenDoc e' promosso da IBM. Esso e' basato su CORBA. Ne esistono gia' implementazioni su Windows, Mac, AIX e OS/2 di IBM.

OpenDoc e' un sistema di oggetti distribuiti basato su CORBA (il suo run-time e' costituito da oggetti CORBA ed usa il linguaggio di dispatching di SOM), piu' avanzato da molti punti di vista rispetto a COM/DCOM. Esso supporta l'integrazione di componenti dentro altre applicazioni, come fa OLE di proprieta' Microsoft. Esso e' promosso da CIL (Component Integration Labs) di cui IBM e' uno dei maggiori membri).

Esso e' basato su un sistema ad oggetti tipo SOM/DSOM di IBM.

In questi giorni si sta parlando di un abbandono di questa tecnologia in favore di JavaBeans.

IMPLEMENTAZIONI DI CORBA

Implementazioni "free" di CORBA:

ILU di Xerox Park

Electra

DOME: commerciale, bimario free per Linux

CHORUS/COOL ORB: commerciale, binario disponibile per valutazione

PONTE TRA I DUE SISTEMI CORBA E DCOM

Esistono tecnologie di ponte tra i due standard di oggetti distribuiti.

L'OMG (Object Management Group) ha definito un ponte tra CORBA e COM.

ARCHITETTURA RMI

RMI (Remote Method Invocation) permtte il passaggio trasparente di messaggi tra oggetti che possono risiedere su macchine e sistemi operativi diversi.

E' un metodo che si utilizza per oggetti remoti Java in rete, i cui metodi possono essere invocati da una Virtual Machine diversa.

Esempi di RMI sono l'HORB e l'RMI di Java (JavaSoft). Non sempre e' facile inviluppare le applicazioni "legacy" in oggetti Java, specialmente le applicazioni server.

Lo svantaggio dell'RMI rispetto a CORBA e' che non supporta molteplici linguaggi di programmazione.

TECNOLOGIE DI COMPONENTI PER IL WEB

JAVABEANS ED ACTIVEX

JavaBeans di JavaSoft (SUN) ed ActiveX di Microsoft sono due "framework" a componenti, che, pur essendo diversi in "scope", sono oggetto di una dura battaglia.

I punti di vista sotto i quali essi possono essere confrontati sono: disponibilita', capacita', sicurezza, interoperativita' e indipendenza dalla piattaforma.

JAVABEANS

JavaBeans e' una tecnologia che Sun ed IBM stanno sviluppando per dare alle applicazioni Java la funzionalita' di documento composto che gia' hanno OpenDoc e ActiveX.

JavaBeans e' un modello di componenti per costruire ed usare componenti scritti in Java. JavaBeans e' una specifica pubblica sviluppata da Sun in collaborazione con altri attori Java.

Un bean e' una classe Java con qualche informazione descrittiva in piu'.

JavaBeans e' un'architettura a componenti sviluppata da JavaSoff (SUN) ed IBM per dare ad applicazioni Java una funzionalita' di documenti composti (software a componenti) tipo OpenDoc ed AciveX. Cosi' come anche ActiveX, e' una tecnologia mirata ad applicazioni basate sul Web.

SunSoft afferma che JavaBeans liberera' una volta per tutte gli sviluppatori dai vincoli delle diverse piattaforme.

La sicurezza (vista sia come safety sia come security) dei JavaBeans fa affidamento a quella propria del linguaggio Java. Attualmente ci sono le applicazioni Java stand-alone che hanno accesso illimitato alle risorse del computer su cui vengono eseguite localmente e le cosiddette applet (applicazioni provenienti al computer via rete da un server) che hanno un accesso severamente ristretto. Il nuovo JDK (Java Development Kit) introduce le applet firmate, cioe' applet provenienti da una sorgente fidata, che permettono un accesso alle risorse esteso, in seguito al controllo della firma. Una parte della sicurezza e' poi affidata ai browser.

JavaBeans e' cross-platform: La Java Virtual Machine (interprete Java) e' infatti disponibile sui piu' diffusi sistemi operativi.

Microsoft sta predisponendo l'utilizzazione dei JavaBeans come controlli ActiveX.

JavaBeans ha posto in prima piano il problema dell'interoperabilita' con gli altri sistemi a componenti attraverso dei ponti.

ARCHITETTURA VISIBROKER

VisiBroker di Visigenic.

ARCHITETTURA HORB (CAT)

ARCHITETTURA NEO (SUN)

OPENSTEP

OpenStep e' la specifica di API (Application Programming Interface) scritta per i sistemi NeXT, ma indipendente dal sistema operativo. In esso sono compresi gli oggetti distribuiti.

GNUSTEP (OpenStep)

In GNUstep, sistema ad oggetti comprendente il supporto ad oggetti distribuiti, sono state "portate" molte delle funzionalita' di OpenStep

FRESCO

Fresco di FASlabs e' un sistema "free" che raggiunge molti degli obbiettivi di OpenDoc ed OLE integrando diverse parti in una "pagina" di applicazione. E' basato su CORBA ed ha alcune caratteristiche in piu' di OpenDoc, am e' a piu' basso livello. Sembra che non sara' piu' sopportato.

JOE di SunSoft

permette l'accesso di applet Java a server basati sull'ORB Java di Sun NEO (un implementazione CORBA).

IIOP

Netscape ha inserito nel suo nuovo browser (Communicator) un Internet Interoperable ORB Protocol (IIOP).